

## 1 Logikrätsel

(4 Punkte)

Aus *Norvig/Russell: AIMA 3rd ed.* stammt folgendes Rätsel:

In fünf Häusern, die in unterschiedlichen Farben angestrichen sind, leben fünf Personen unterschiedlicher Nationalitäten, die jeweils unterschiedliche Zigarettenmarken rauchen, ein jeweils unterschiedliches Getränk bevorzugen und jeweils unterschiedliche Haustiere haben. In jedem Haus lebt eine Person.

Folgende Fakten sind bekannt:

- Der Engländer lebt im roten Haus.
- Dem Spanier gehört der Hund.
- Der Norweger lebt im ersten Haus links.
- Im gelben Haus werden Kools geraucht.
- Der Mann, der Chesterfields raucht, lebt im Haus neben dem des Mannes mit dem Fisch.
- Der Norweger lebt neben dem blauen Haus.
- Der Winston-Raucher hält Schnecken.
- Der Lucky-Strike-Raucher trinkt Orangensaft.
- Der Ukrainer trinkt Tee.
- Der Japaner raucht Parliaments.
- Im Haus neben dem Haus mit dem Pferd werden Kools geraucht.
- Im grünen Haus wird Kaffee getrunken.
- Das grüne Haus ist unmittelbar rechts vom elfenbeinfarbenen Haus (von Ihnen aus gesehen).
- Im mittleren Haus wird Milch getrunken.

Formulieren Sie das Problem als CSP (Variablen, Wertebereiche, Constraints) auf dem Papier.

*Hinweis:* Machen Sie sich zunächst klar, was die Variablen und was deren Wertebereiche sind. Schreiben Sie die Constraints als (unäre bzw. binäre) Relationen auf.

**Thema:** Formulierung von Problemen als CSP

## 2 Framework für Constraint Satisfaction

(2 Punkte)

Checken Sie das AIMA-Repository<sup>1</sup> aus. Im Paket `aima.core.search.csp` finden Sie Java-Klassen zum Umgang mit CSPs sowie einige der in der VL besprochenen Algorithmen und Heuristiken.

Nutzen Sie das Framework, um eine Lösung für das Problem aus der vorigen Aufgabe zu finden.<sup>2</sup> Beantworten Sie folgende Frage:

Wo lebt das Zebra und welchem Haus wird Wasser getrunken?

*Hinweis:* Sie können den Solver auch in einer Sprache Ihrer Wahl selbst implementieren.

**Thema:** Implementierung und Anwendung von CSP-Algorithmen

---

<sup>1</sup>Sie finden das Repository unter [github.com/aimacode/aima-java](https://github.com/aimacode/aima-java).

<sup>2</sup>Falls Sie die vorige Aufgabe nicht gelöst haben, können Sie hier gern ein anderes Problem modellieren und lösen.

### 3 Forward Checking und Kantenkonsistenz

(4 Punkte)

Betrachten Sie das 6-Damen Problem, bei dem 6 Spielfiguren auf einem  $6 \times 6$  Felder großen Brett so platziert werden sollen, dass sich keine zwei Damen auf der selben horizontalen, vertikalen oder diagonalen Line befinden. Die 6 Spalten werden als Variablen  $v_i$  mit  $i = 1, \dots, 6$  modelliert. Der Wertebereich sei  $D_{v_i} = 1, \dots, 6$  für alle Variablen  $v_i \in V$ , d.h. der Wert einer Variablen gibt die Zeile an, in der eine Spielfigur in der entsprechenden Spalte platziert ist. *Beispiel:*  $v_2 = 5$  bedeutet, dass in Spalte 2 und Zeile 5 eine Dame platziert ist.

Betrachten Sie nun die Zuweisung  $\alpha = \{v_1 \rightarrow 2, v_2 \rightarrow 4\}$ .

- a) Erzeugen Sie Kantenkonsistenz in  $\alpha$ . Geben Sie hierzu die Wertebereiche der Variablen vor und nach dem Erzeugen der Kantenkonsistenz an.

*Hinweis:* Sie dürfen annehmen, dass der Wertebereich von Variablen mit bereits zugewiesenen Werten nur aus dem zugewiesenen Wert besteht, während unbelegte Variablen den vollen Wertebereich haben.

- b) Führen Sie Forward-Checking in  $\alpha$  aus. Vergleichen Sie das Ergebnis mit (a).

**Thema:** Kantenkonsistenz und Forward Checking verstehen

### 4 Kantenkonsistenz mit AC-3

(4 Punkte)

Sei  $D = \{0, \dots, 5\}$ , und ein Constraintproblem definiert durch  $\langle (v_1, v_2, v_3), (D, D, D), \{c_1, c_2, c_3\} \rangle$  mit

- $c_1 = ((v_1, v_2), \{(x, y) \in D^2 \mid x + y = 3\})$ ,
- $c_2 = ((v_2, v_3), \{(x, y) \in D^2 \mid x + y \leq 3\})$  und
- $c_3 = ((v_1, v_3), \{(x, y) \in D^2 \mid x \leq y\})$ .

Wenden Sie den AC-3-Algorithmus auf das CSP an. Geben Sie den Zustand der Queue und das Ergebnis von *Revise*, d.h. den Ergebniszustand des aktuellen  $D_i$ , für jede Iteration des Algorithmus an.

**Thema:** Handsimulation des AC-3-Algorithmus

### 5 Anwendungen

(2 Punkte)

Recherchieren Sie, in welchen Anwendungen CSP vorkommen und mit der BT-Suche (plus Heuristiken) oder sogar AC-3 gelöst werden. Erklären Sie kurz, wie und wofür die Algorithmen jeweils genutzt werden.

**Thema:** Anwendungen von CSP, BT-Suche und AC-3